ANÁLISE DE DADOS: UM COMPARATIVO ENTRE A LINGUAGEM R E PYTHON

Ulisses Dias de Lima Arêas[1], Fabrício Gustavo Henrique[1]

[1]Faculdade de Tecnologia de Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

ulisses.areas@fatec.sp.gov.br,
fabricio.henrique@fatec.sp.gov.br

Resumo. *Este artigo analisa a utilização da linguagem Python e da linguagem R para o fim de realização de análise de dados. O objetivo é verificar se existem vantagens, além da preferência pessoal, na utilização de uma dessas linguagens, considerando suas características e capacidades. A pesquisa é baseada em análise bibliográfica e documental. Pesquisas mostram que a linguagem Python se destaca como a mais popular, seguida do SQL e em terceiro lugar da linguagem R e essa preferência pelo Python se dá pela capacidade da linguagem de atender as necessidades dos usuários, que lidam com problemas complexos e necessitam de uma linguagem capaz de prover o ferramental necessário para efetivamente lidar com as quatro grandes etapas que compõem o processo analítico de dados: coleta e limpeza, exploração, modelagem e visualização.*

Abstract. *This article analyses the use of both Python and R programming languages for the purpose of data analysis. The objective is to verify the existence of advantages, beyond personal preferences, in the use of one of this two languages, considering its characteristics and capabilities. The research is based on bibliographic and documentary analysis. Researches shows that Python is the most popular programming language followed by SQL and R, and this preference comes from the capacity of Python to meet the user's needs when dealing with complex problem solving situations that requires a programming language capable of providing the necessary tools to cope with the four major steps in data analytics: data collection and cleaning, data exploration, data modeling and data visualization.*

1. Introdução

Estatísticos, analistas e cientistas de dados, estudantes e membros de outras comunidades necessitam, com razoável frequência, lidar com dados que, se não tratados, podem não apresentar de maneira tão clara as informações que contêm. Dados são objetos brutos, sem informação explícita uma vez que precisam ser processados para que algo útil possa ser extraído. Informação é o resultado da análise dos dados após a submissão dos mesmos a um processo que ocorre em diferentes níveis e fases. Não é fácil o processamento manual de dados e por essa razão os usuários recorrem a ferramentas que auxiliam na extração das informações intrínsecas.

A quantidade de dados disponível cresce diariamente a uma razão vertiginosa.

Segundo uma pesquisa da Finances Online Reviews for Business, em 2021 o volume de dados criado diariamente foi de 1.134 trilhões MB, enquanto estima-se um consumo de 118 Zettabytes (118 x 10^{21} Mb) para o ano de 2023. Para efeito de comparação, o consumo no ano de 2021 foi de 74 ZB e em 2022, 94 ZB. Uma possível explicação para esse aumento no consumo de dados está na forma como os negócios e organizações valem-se cada vez mais desse ativo para as tomadas de decisão, denominadas *data-driven*, ou seja, decisões baseadas em dados.

A análise de dados é simplesmente a análise de uma coletânea de dados submetidos a um processo de limpeza, transformação em algo compreensível e modelagem para extração de informações de interesse. Para a realização desses processos o usuário necessita de ferramentas adequadas, e o foco deste trabalho está nas facilidades (bibliotecas e ferramentas) das linguagens R e Python, suas interseccionalidades e diferenças no tratamento de dados estruturados, e não na apresentação de metodologias de análise de dados. Não é intenção, consequentemente, classificar tais linguagens e sim trazer um pouco de clareza sobre as possibilidades apresentadas por cada uma para auxiliar aquele usuário iniciante que deseja se aventurar no mundo da análise de dados.

A metodologia utilizada neste trabalho foi a pesquisa bibliográfica e documental.

Este trabalho é dividido em 5 seções. A primeira, Introdução, traça um breve panorama sobre o que é análise de dados e as linguagens R e Python como ferramentas de auxílio à atividade. A segunda, terceira e quarta seções discorrem sobre o apoio teórico que embasa este trabalho e discute, em termos mais detalhados, características das linguagens R e Python e a interoperabilidade possível entre elas. Por fim, a última seção traz as considerações finais resultado das discussões apresentadas.

2. A linguagem R

A linguagem R foi desenvolvida para ser uma linguagem de computação para análises estatísticas e é fortemente influenciada por duas linguagens predecessoras: S e Scheme. Segundo seus criadores, a linguagem foi desenvolvida de forma que sua sintaxe se assemelhe à linguagem S e a semântica à Scheme.

Uma linguagem de programação é descrita em termos de sintaxe e semântica. O que é possível ou não ser realizado com a linguagem é determinado pela semântica enquanto, por outro lado, a sintaxe é importante porque determina o modo como o usuário consegue se expressar através daquela linguagem (IHAKA; GENTLEMAN, 1996). Contudo, sintaxe e semântica por si só não produzem uma linguagem útil, elas precisam ser combinadas com outros fatores, sendo um deles as ferramentas disponíveis (*tools*) em cada linguagem, de forma que, neste contexto, um bom ambiente em tempo de execução esteja disponível (IHAKA; GENTLEMAN. 1996).

Para o desenvolvimento da linguagem R foi escolhido o modelo subjacente de avaliação da linguagem Scheme, a semântica, junto com a sintaxe da linguagem S. A escolha da sintaxe da linguagem S se deu em virtude de esta representar, para os analistas de dados, um poderoso meio de expressar os cálculos computacionais que eles necessitavam conduzir (IHAKA; GENTLEMAN. 1996). Para Tierney (1990), o continuado sucesso do S relativo ao Lisp-Stat, linguagem de domínio específico para análise estatística e aprendizado de máquina voltada para praticantes de análises estatísticas com pouca ou nenhuma experiência em programação, é devido a provisão de

uma estrutura sintática capaz de expressar ideias estatísticas de uma maneira que estatísticos se sintam confortáveis. De fato, a comunidade R Foundation defende que o R é um ambiente dentro do qual técnicas estatísticas são implementadas, e não necessariamente um sistema estatístico.

O Manual da Linguagem R a define como um conjunto integrado usado para manipulação de dados, cálculos e criação de gráficos apresentando facilidades como: efetivo manuseio e guarda de dados; um conjunto de operadores de cálculo, particularmente matrizes; uma coleção integrada, coerente e abrangente de ferramentas para análise de dados; facilidades gráficas para análise de dados e por fim, uma linguagem de programação bem desenvolvida, simples e eficaz. Ainda, que a linguagem é um veículo para novos métodos de desenvolvimento de análise interativa de dados.

R é uma linguagem de programação multi-paradigma orientada a objetos, de programação funcional, dinâmica, fracamente tipada, voltada à manipulação, análise e visualização de dados e utilizada por pesquisadores, estudantes e analistas em diversas áreas tais como, Estatística e Ciências da Computação.

A linguagem R possui diversos pacotes que podem ser utilizados para o fim de análise de dados. Alguns dos mais utilizados são:

RODBC

Esse pacote permite a recuperação de dados diretamente do banco de dados. Com ele é possível se conectar diretamente a tabelas que estão dentro do SGBD (Sistema de Gerenciamento de Banco de Dados) com uso de canais ODBC (*Open Database Connection*) e recuperar dados do sistema, dispensada a utilização de interfaces csv, xlsx ou json. Com essa biblioteca é possível utilizar linguagem de consulta para transformar dados que estão dentro de um SGBD diretamente em um *dataframe*.

Dplyr

Para o processo de *data wrangling*, ou seja, quando se realiza a transformação e manipulação dos dados em outros formatos para extração de informações úteis, o pacote *dplyr* é uma biblioteca de manipulação de dados que por meio do pipeline (%>%) permite a execução dessa tarefa de forma otimizada. Como a sintaxe da linguagem R não é limpa quando se trata de lidar com *pipeline*s de dados complexos, a adição de vários filtros ou mesclagem de *dataframes* pode tornar o código confuso. O poder do *dplyr* está em otimizar e potencializar o processo de manipulação de dados, reduzindo a complexidade da tarefa.

Caret

Dentro da linguagem R, o pacote *caret* é considerado um dos mais completos quando se trata de aprendizado de máquina. O *care*t (*Classification and Regression Training*) possui uma série de funções que agilizam o processo de criação de modelos preditivos, com ferramentas para separação dos dados, pré-processamento, seleção de recursos, ajuste de modelo com uso de reamostragem e estimação da importância da variável, dentre outras. A utilização desse pacote torna simples a comparação do desempenho e dos resultados dos modelos, o que é tarefa padrão em projetos de aprendizado de máquina.

Ggplot2

Biblioteca que permite criar gráficos de todos os tipos, com elevado nível de

customização e qualidade. Com o *ggplot2* os dados são fornecidos e então as variáveis são mapeadas, adequadas a estética necessária e como resultado tem-se gráficos criados declarativamente.

3. A linguagem Python

Python foi criado no início dos anos 1990 por Guido van Rossum como uma linguagem sucessora da ABC. O trabalho continuado deste autor no Corporation for National Research Initiatives (CNRI), em Reston, Virginia, em 1995, resultou no lançamento de diversas versões da linguagem. Em 2001 a Python Software Foundation (PSF) foi criada como uma organização sem fins lucrativos especificamente para ser a proprietária intelectual do Python.

Como ferramenta para análise de dados, a linguagem Python possui um ecossistema de bibliotecas orientadas a dados, como por exemplo, NumPy, Pandas, Seaborn Matplotlib, Scikit-learn, etc, essenciais e complementares na execução das atividades de manipulação, processamento, limpeza e extração de informações de dados (MCKINNEY, 2018). Na última década, dentre as linguagens interpretadas, Python angariou uma ativa comunidade voltada para o processamento científico e de dados, deixando de ser reconhecida apenas como uma linguagem inovadora a ser usada por conta e risco (MCKINNEY, 2018).

Nos últimos anos ocorreu um incremento relacionado às capacidades das bibliotecas do Python, otimizando a linguagem, o que, por um lado e em certa medida justifica a popularidade crescente dela, conforme pode ser observado na Figura 1. Associa-se a isso a possibilidade de utilização da linguagem para construção de aplicações de dados, o que apenas confirma sua robustez.

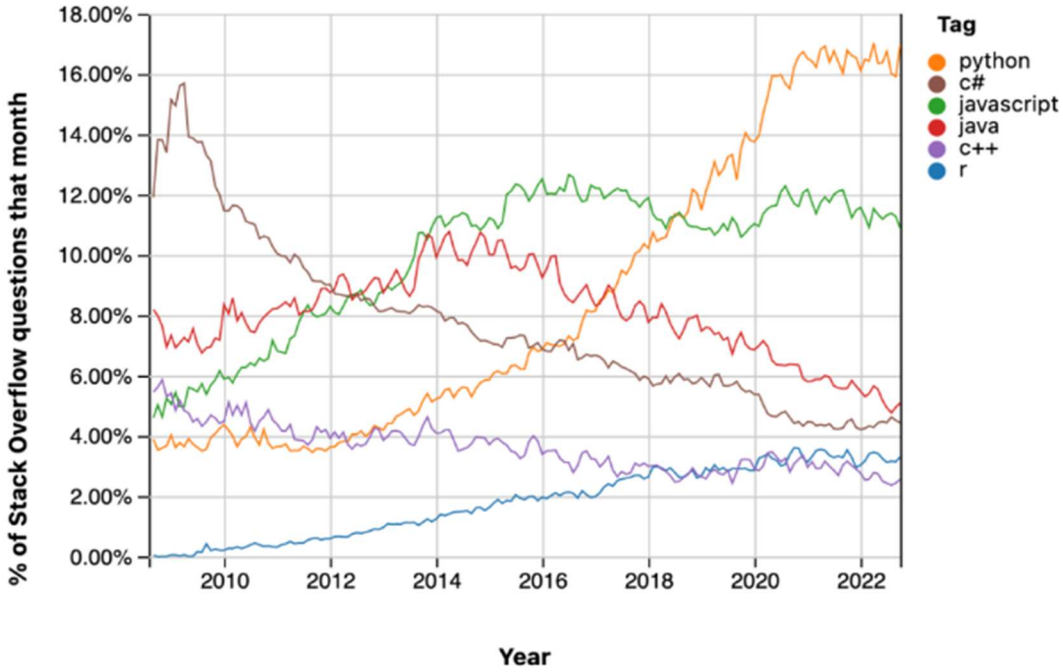

Figura1. Crescimento das principais linguagens de programação

Fonte: (Stack Overflow, 2021

Por outro lado, o sucesso da linguagem pode ser vinculado à capacidade do Python se integrar com códigos em C, C++ e FORTRAN, ainda mais quando se observa que existem ambientes de processamento modernos que compartilham um conjunto semelhante de bibliotecas FORTRAN e C legadas para álgebra linear, otimização, integração e outros algoritmos desses tipos (MCKINNEY, 2018). Há de se destacar que a linguagem Python é apropriada não só para pesquisas e prototipagem, mas também para a construção de sistemas de produção, ou seja, é uma linguagem que consegue atender às necessidades do usuário entregando resultados que outras linguagens mais especializadas entregariam, como SAS e R, permitindo que se transporte as ideias para o campo de produção, que, em tese, seria escrito em Java, C# ou C++. Como destaca McKinney (2018) com isso resolve-se o problema do uso de duas linguagens, mantendo-se apenas um ambiente de produção.

McKinney (2018) destaca que ainda que Python seja um ambiente propício e adequado para a construção de vários tipos de aplicações de análise e sistemas de propósito geral, existem usos para os quais a linguagem pode não ser tão apropriada e isso pode ser associado ao fato de que o Python, por ser uma linguagem interpretada, geralmente a maior parte do código executará de forma consideravelmente mais lenta do que um código escrito em linguagem compilada, como Java ou C++.

Conforme mencionado anteriormente, a linguagem Python possui um ecossistema de bibliotecas orientadas a dados, e as características de algumas dessas bibliotecas será apresentada a seguir.

NumPy

Essa biblioteca oferece vários recursos para cálculos numéricos e científicos, permitindo que o usuário trabalhe com matrizes de alto desempenho que fornecem vetorização de operações matemáticas. Os objetos DataFrame e Panda series dependem de matrizes NumPy para realização dos cálculos matemáticos. De acordo com McKinney (2018), a NumPy tem sido a pedra fundamental do processamento numérico em Python por oferecer o código aglutinador para as estruturas de dados, os algoritmos e a biblioteca necessários à maioria das aplicações científicas envolvendo dados numéricos. A Numpy é utilizada, também, como um contêiner para que os dados sejam passados entre algoritmos e bibliotecas.

Pandas

O pandas permite o trabalho com dados estruturados ou tabulares de forma rápida, fácil e expressiva, tendo como principais objetos o *DataFrame* e as Series, que é um objeto array unidimensional (MCKINNEY, 2018). Ainda, esta biblioteca disponibiliza funcionalidades de indexação que facilitam a formatação, manipulação, as agregações e a seleção de subconjuntos de dados, caracterizando-se como uma das principais ferramentas da linguagem.

Matplotlib

Desenvolvida para a criação de diagramas, gráficos bidimensionais e plotagens apropriadas para publicação, sendo a mais amplamente utilizada ainda que existam outras

bibliotecas de visualização disponíveis.

Seaborn

O seaborn é uma biblioteca baseada na matplotlib que contém configurações padrão mais adequadas para o processamento gráfico e ajuda o usuário a explorar e entender os dados. Suas funções operam nos *dataframes* e arrays que contém o conjunto de dados e internamente desempenha o mapeamento semântico e a agregação estatística necessários para produção de plotagens informativas. Seaborn utiliza o matplotlib para compor as plotagens, mas se a intenção são gráficos interativos, é recomendável utilizar a interface Jupyter/IPyhton no modo matplotlib.

Scikit-learn

Modulo Python baseado no Numpy e SciPy que fornece algoritmos para tarefas padrão de aprendizado de máquina e mineração de dados, como clustering, regressão, classificação, redução de dimensionalidade e seleção de modelo (MCKINNEY, 2018). Juntamente com o Pandas, Statsmodels e IPython, essa biblioteca tem sido crucial para possibilitar que o Python seja uma linguagem de programação produtiva para a ciência de dados.

Statsmodels

O foco maior deste pacote é em inferência estatística, oferecendo estimativas de incerteza e valores *p* para parâmetros. Em oposição, o scikit-learn está mais centrado em previsões.

4. Interoperabilidade linguística

Interoperabilidade é a capacidade de diferentes linguagens de programação trabalharem conjuntamente, e isso é um pilar em computação, embora possa ser problemático por diversas razões, por exemplo, a utilização de memória e a incompatibilidade de estruturas de armazenamento de dados (SCAVETTA, 2021). Embora há tempos se tente efetuar uma implementação desse tipo entre o Python e o R, apenas nos últimos anos algo razoavelmente funcional foi concretizado.

Na prática, se o usuário do R necessita acessar o Python, então o pacote *reticulate* é adequado. Esse pacote fornece um conjunto de ferramentas para interoperabilidade entre o Python e o R, incluindo recursos para importação de módulos Python e uso interativo da linguagem dentro de uma sessão R. Ainda, permite a tradução entre objetos R e Python além de ser uma ligação flexível para diferentes versões do Python, incluindo os ambientes virtuais. Por outro lado, se primariamente se utiliza o Python, então a biblioteca *rpy2* é a ferramenta indicada. Ela é uma interface do R através do Python, legada da RSPython, de 2005, que após melhoramentos gerou a RPy até chegar, posteriormente, à RPy2. A figura 2 apresenta comandos em linguagem R e Python vinculados ao pacote *reticulate* e à biblioteca *rpy2* como exemplo do que foi apresentado nas linhas acima.

Table 6-1. Interoperability granted by reticulate

Access	Using command
Python Functions	in R, `pd <- library(pandas); pd$read_csv()`
Python Objects	in R, `py$objName`
R Objects	in Python, `r.objName`

Table 6-2. Interoperability granted by rpy2 when writing in Python

Access	Using command
R Functions	in Python, `import rpy2.robjects.lib.ggplot2 as ggplot2`
R Packages	in Python, `r_cluster = importr('cluster')`
R Objects	in Python, `foo_py = robjects.r['foo_r']`

Figura 2. Tabelas com os comandos para acessar objetos de uma linguagem para outra

Fonte: (Scavetta, 2021)

Quando se considera apenas a funcionalidade, o R é mais voltado para a comunidade estatística em si, enquanto o Python sagra-se como uma linguagem mais universal, e um indicador disso é o crescimento no uso das principais linguagens de programação no mundo.

5. Considerações Finais

É inegável o sucesso da linguagem Python atualmente, o que pode ser verificado nas pesquisas de popularidade existentes. Quando se analisa o histórico da linguagem, observamos que sua existência remonta aos anos 1990, ou seja, a linguagem teve seu tempo para amadurecer e conquistar uma comunidade de apoio. Dada sua popularidade, é alta a probabilidade de que se encontre a solução de qualquer problema que se enfrente com o uso da linguagem por meio de uma rápida busca no Google. A existência de tutoriais explicativos aliados a uma sintaxe facilmente legível para humanos, são algumas das facilidades que torna o Python uma linguagem extremamente amigável para iniciantes. É preciso destacar sua versatilidade, onde para cada necessidade há, ao menos, um pacote associado, ou seja, para qualquer tarefa requerida, é provável que exista um pacote para atendê-la. Por outro lado, como toda linguagem, há algumas desvantagens no uso do Python. Por exemplo, o script Python não é primeiro compilado e então executado; em vez disso, ocorre a compilação todas as vezes que o script é executado, portanto, os erros de codificação se manifestam no tempo de execução (*runtime*). Isso faz com que haja uma redução do desempenho da linguagem. Contudo, para iniciantes isso pode ser considerado algo bom, pois se aprende com os erros, pois todas as vezes que um problema desses ocorre, vários testes têm que ser feitos para se encontrar a solução. Quanto à velocidade e execução, o Python costuma ser mais lento que outras linguagens de programação, ainda que, nos últimos anos, tenha ocorrido uma melhoria neste aspecto. Em resumo, o Python deve continuar em alta, mas há questões relativas à linguagem que precisam ser melhoradas e outras linguagens já o estão fazendo.

Quanto ao R, a linguagem sempre foi referência no campo de análise de dados,

tendo sido, fundamentalmente, criada para esse fim. Contudo, entre 2016/2017, a utilização do Python superou-a, conforme ilustrado na Figura 3.

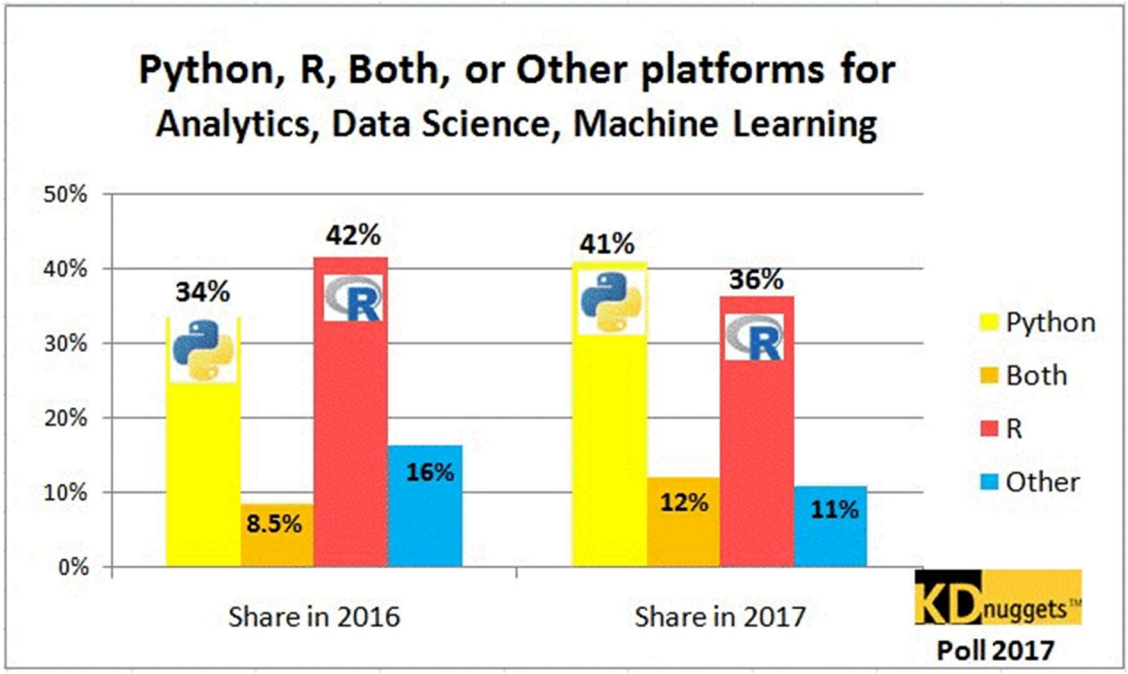

Figura 3. Quota do usos das linguagens R e Python em Data Science

Fonte: (KDnuggets, 2017)

A partir de então o crescimento da preferência pela linguagem Python se tornou cada vez mais aguda, contudo, isso não deve ser interpretado como uma superioridade do Python em relação ao R. Quando se leva em consideração os conceitos básicos e as necessidades da área de ciência de dados, é possível afirmar que tudo o que uma linguagem consegue executar a outra também consegue. Tanto as funções que compõem o R base quanto as oferecidas pelo Tidyverse permitem ao usuário, de forma muito simples e direta, tratar, manipular e visualizar os dados. Além das bibliotecas que compartilham similaridades com as funções encontrados em Python, há outras que não concorrem com àquela linguagem, a exemplo da estrutura de modelagem estatística GAMLSS (*Generalized Additive Models for Location, Scale and Shape*), modelos de regressão distributiva univariados, onde todos os parâmetros da distribuição assumida para a resposta podem ser modelados como funções aditivas das variáveis.

O que foi apresentado neste artigo não intentou esgotar todo o arcabouço de possibilidades, mas sim apresentar algumas características das linguagens R e Python e mostrar que, independentemente da linguagem e das ferramentas utilizadas, mais importante é o conhecimento subjacente da estatística e dos modelos que se pretende usar bem como do problema que se pretende resolver, pois esses são o cerne do processo de análise de dados. A escolha de qual linguagem será utilizada acaba por atender a alguns

critérios de preferência pessoal e de necessidade imposta pelo que se pretende analisar. O olhar que deve ser lançado sobre a questão R versus Python é, primordialmente, se desprender de qualquer rivalidade e rumar para a utilização conjunta das linguagens. Nesse contexto, já existem projetos que propõem a utilização de um ambiente de desenvolvimento integrado que comporta a codificação em R e Python, caso do projeto *prython*, uma nova IDE para o Windows que permite ao usuário codificar em R ou Python por meio da interconexão de painéis em uma tela. Esse é apenas um exemplo dos esforços que estão sendo feitos para se extrair o máximo de proveito dessas duas linguagens em prol da ciência de dados.

Referências

ANGELOV, B. e SCAVETTA, R.J. (2021) Python and R for the modern data scientist: The best of both worlds. First edition. Sebastopol: Oreilly, 2021.

BHAVSAR, P.D. (2020) Why Python is one of the most preferred languages for data scientists. Disponível em:< https://www.kdnuggets.com/2020/01/python-preferred-languages-data-science.html>. Acesso em: 08/04/2023.

CANTARATO, V. (2020) R2Py. Disponível em:<https://contaratoandressa.medium.com/r2py-1120f9def650>. Acesso em: 01/05/2023.

FOUNDATION, The R. (2023) What is R? Disponível em:< https://www.r-project.org/about.html>. Acesso em 09/04/2023.

GENTLEMAN, R. e IHAKA, Ross. (1996) A Language for Data Analysis and Graphics. Journal of Computational and Graphical Statistics, Volume 5, Number 3, Pages 299-314.

GOMES, F. (2018) Reticulate. Disponível em: https://rpubs.com/gomes555/reticulate. Acesso em: 01/05/2023

GROLEMUND, G. e WICKHAM, H. (2019) R para Data Science. 1ª ed. Rio de Janeiro: Alta Books, 2019.

GRUS, J. (2016) Data Science do Zero: Primeiras Regras com o Python. 1ª ed. Rio de Janeiro: Alta Books, 2016.

JOURY, A. (2023) O Python é muito útil, mas pode não ser a linguagem de programação do futuro. Disponível em:< https://tecnoblog.net/especiais/o-python-e-muito-util-mas-pode-nao-ser-a-linguagem-de-programacao-do-futuro/>. Acesso em: 01/05/2023

LOUI, A. (2023) 53 important statistics about how much data is created every day. Disponível em:< https://financesonline.com/how-much-data-is-created-every-day/>. Acesso em: 08/04/2023

MATOS, D. (2022) 5 melhores pacotes da linguagem R para cientistas de dados. Disponível em:< https://www.cienciaedados.com/5-melhores-pacotes-da-linguagem-r-para-cientistas-de-dados/>. Acesso em: 01/05/2023

MAYO, M. (2020) R or Python? Why not both?. Disponível em:< https://www.kdnuggets.com/2020/12/r-python-both-prython.html>. Acesso em: 09/04/2023.

MCKINNEY, W. (2018) Python para análise de dados. 1ª ed. São Paulo: Novatec Editora, 2018.

ROCHA, R. (2020) R vs Python – uma análise desapaixonada. Disponível em:< https://www.flai.com.br/ricardo/r-vs-python-uma-analise-desapaixonada/#:~:text=Se%20voc%C3%AA%20est%C3%A1%20come%C3%A7ando%20e,escolher%20s%C3%B3%20uma%2C%20escolha%20Python>. Acesso em: 08/04/2023.